AISG Test Tool (Secondary Demo mode) Installation and User Guide

Revision History

DATE	ISSUE	NOTES
2004.02.24	1.0	Document First release – M. Lopez
2004.04.15	1.1	Release for Secondary Mode Demo version

TABLE OF CONTENTS

1. REQUIREMENTS

1.1 Hardware

- 1 serial port (2 if Interface will be run as Control station and Virtual device in the same machine)
- 1 Null modem cable (RS232).
- Serial Connection parameters supported are: 9600, 38400 and 115200 Kbps, no flow control, 8 bits data, no parity, 1 stop bit.
- Video Graphics Adapter – at least 256 colour palette (Java requirement)
- Hard Disk space – at least 50 MB for operation (depends on application log size), 20 MB in Windows root directory for temporary files during installation.
- *Software was tested on PC's with Pentium III @ 800 MHz, 256 Mb RAM and Pentium Celeron @ 1.3GHz, 256 Mb RAM*

1.2 Software

- Windows 98/NT4.0_SP6/2000/XP.
- Java™ 2 Runtime Environment, Standard Edition, build 1.4.1_02-b06 or higher.
- Java HotSpot™ Client VM, build 1.4.1_02-b06, mixed mode.
- Java™ Communications API version 2.0.
- *Tested on Windows 98, Windows 2000 SP2 and Windows XP SP1.*

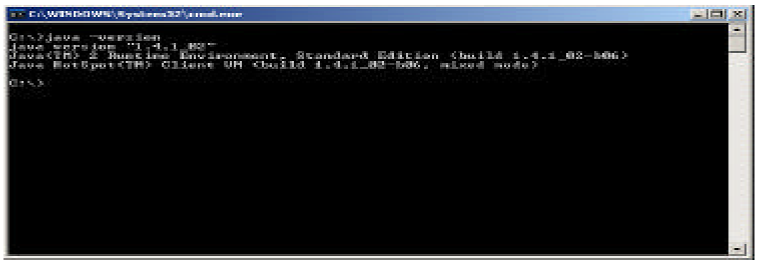
REQUIRED VERSIONS OF
JAVA COMPONENTS ARE NOT SUPPORTED IN WINDOWS 95

2. INSTALLATION

2.1 Java Runtime Environment (JRE) manual installation

If a version older than 1.4.1_02 is installed in the computer, it is strongly recommended to upgrade it. If more than one JRE is installed, make sure that the running version is at least 1.4.1_02.

1. Verify the Java version installed in the computer. From a Command prompt window type the following command: ***java –version***



```
C:\WINDOWS\System32\cmd.exe
C:\>java -version
java version "1.4.1_02"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.1_02-b06)
Java HotSpot(TM) Client VM (build 1.4.1_02-b06, mixed mode)

C:\>
```

Figure 1. *Java Runtime Environment version check*

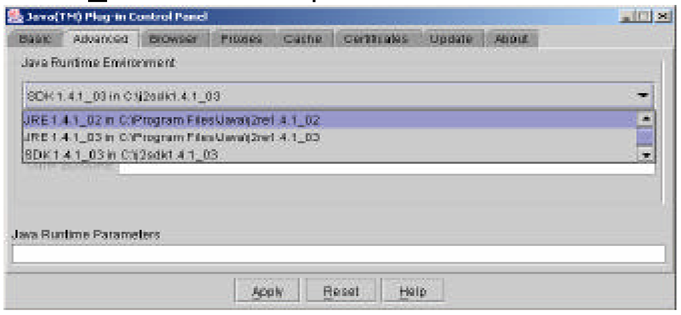
   If the version is 1.4.1_02 or higher, go to next section (installation of *Java Communications API)*.

2. If the available Java version is older than the required one or no JRE is installed, use the JRE installer file provided in the University of Sheffield AISG website (http://www.shef.ac.uk/eee/cr/, ***3rd Party Software, j2re-1_4_1_02-windows-i586-i.exe***). Alternatively, download the latest version from the Java site (http://www.java.sun.com) and follow the on-screen instructions for a typical install.

2.2 Java Communications API (JCA) manual installation

1. Locate the directory where the JRE is installed. By default the JRE installs to the folder: ***<Root>:\Program Files\Java\j2re1.4.1_02***
   (e.g. C:\Program Files\Java\j2re1.4.1_02)

   Confirm the location by opening the Windows Control Panel, then the Java Plug-in configuration panel. Click on the Advanced Properties tab and look for the JRE 1.4.1_02 installation path.



Java Runtime Environment
SDK 1.4.1_03 in C:\j2sdk1.4.1_03
JRE 1.4.1_02 in C:\Program Files\Java\j2re1.4.1_02
JRE 1.4.1_03 in C:\Program Files\Java\j2re1.4.1_03
SDK 1.4.1_03 in C:\j2sdk1.4.1_03
Java Runtime Parameters

Figure 2. *Java Plug-in configuration panel, verification of JRE installation path.*

2. Copy the Java Communications API files to the destinations in the JRE installation path as listed in table 1. The JCA files are provided in the University of Sheffield website (http://www.shef.ac.uk/eee/cr/, ***3rd party software, javacomm20-win32-essentialfiles.zip***).

   *Alternatively, download the files from the Java homepage. The correct version is JCA 2.0. Download the file **javacomm20-win32.zip** and extract only the relevant files*.

File	**Destination**	**Example for the path shown in figure 2** *(modify according to the JRE installation on the PC)*
javax.comm.properties	\lib	C:\Program Files\Java\j2re1.4.1_02\lib
comm.jar	\lib\ext	C:\Program Files\Java\j2re1.4.1_02\lib\ext
win32com.dll	\bin	C:\Program Files\Java\j2re1.4.1_02\bin

Table 1. *Installation files for JCA v. 2.0*

2.1 AISG Test Tool manual installation

1. Create an installation directory for the application and copy into it the files ***aisgApp.jar***, ***launcher.bat, aisg.ico*** and ***guilookproperties.properties.*** *(C:\SheffieldU_AISGTool).*

   Those files are provided in the file ***AISGTestTool1_1sd.zip.***

2. The application requires a path hierarchy to store its related files:
   - <InstallationRoot> - aisgApp.jar, launcher.bat, aisg.ico and guilookproperties.properties must be kept in this location.
   - <InstallationRoot>\devDescriptions – Default location for Device Description Files. *(In the example, C:\SheffieldU_AISGTool\devDescriptions)*
   - <InstallationRoot>\logFiles – This path must be present for the application to run without problems. If not present, it will be created at launch time.
     (In the example, C:\SheffieldU_AISGTool\logFiles)

3. To launch the application, double click the file "***launcher.bat***". This will automatically verify the required tree hierarchy.

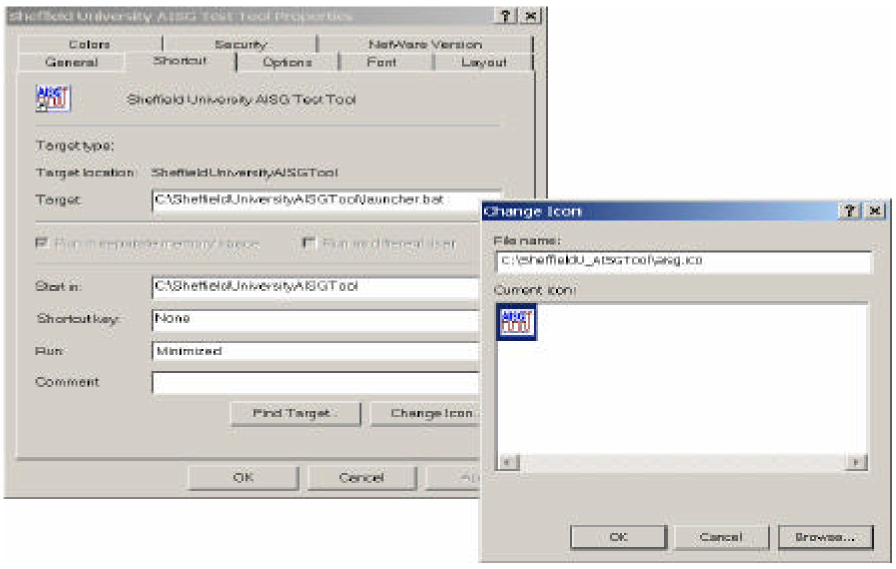
A pre-configured shortcut to launch the application is included in the delivery. The installation path assumed in this shortcut is: ***C:\SheffieldU_AISGTool***

The icon for the pre-configured shortcut might not be displayed correctly on Windows 98, but it still can be used to access the application.

2.2 Modification of the shortcut parameters

1. Open the shortcut properties by right clicking on the shortcut icon and select Properties.
2. Set the correct location of the target file (***launcher.bat***), the start-in directory (*same as the installation directory*) and the icon file location (***aisg.ico***).

The Java application is run using the interpreter from the JRE. This is done through a DOS Command prompt window. Shall a Java exception occur, it will be displayed in this window.



Sheffield University AISG Test Tool

Target type:
Target location: SheffieldUniversityAISGTool
Target: C:\SheffieldUniversityAISGTool\launcher.bat
Start in: C:\SheffieldUniversityAISGTool
Shortcut key: None
Run: Minimized
Comment:

File name: C:\SheffieldU_AISGTool\aisg.ico
Current icon:

Figure 3. *Windows shortcut Properties configuration*

To launch the Java program, a DOS Command prompt window calling the Java interpreter is necessary. When launched using the shortcut, this DOS window is minimised automatically.

DO NOT "Select" text from the DOS window unless the Java application has already been stopped.
Doing so while the application is running will stop all Java execution threads as long as the DOS window is in Select text mode.

3. Application Mode Selection

The version provided as “secondary demo software” does not allow mode selection. Its objective is to provide only a virtual AISG 1.0 Compliant secondary device (either RET, RET type 3, TMA or TMB). This tool allows verification data field formatting, HDLC – AISG frame formatting, link management and AISG global mandatory command set.

Additionally, this version of the software provides an implementation of the optional baud-rate change procedure, log of link activity and allows the user to save/load a device state from a configuration file.

Specific device commands (RET/TMA commands) and the software download protocol implementation are not provided in this version of the software. Please contact us to obtain a full implementation of the software.

The device emulator can hold the description data of only one device at a time, and the represented device can not be deleted. It is possible to overwrite a device, but devices can not have the same Unique ID.

The scope of this application is to test whether the communication protocol specified by AISG1 is followed by the connected devices/primary stations, and the correct data formatting of certain data fields (e.g. tilt value, gain value, etc). A log of the HDLC frame exchange is provided on screen and in the form of a text log located in the ***logFiles*** directory of the application.

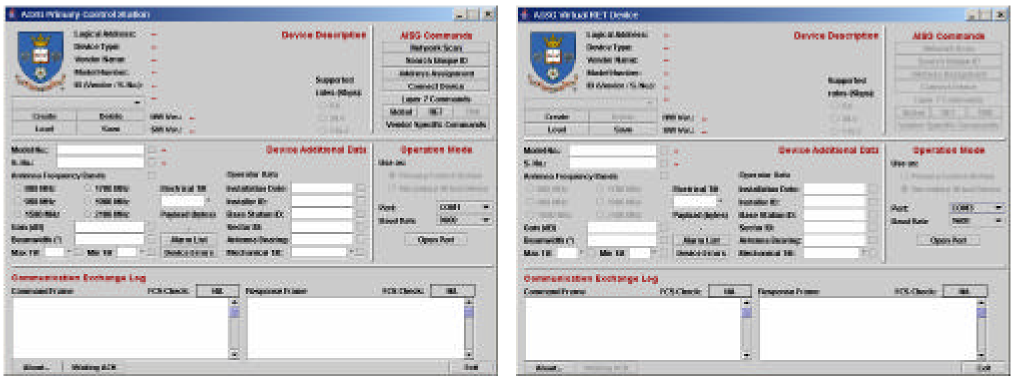


***Figure 4**. Main interface as Primary Station and as Secondary Device. Notice that when used as secondary device, the interface prevents access to management components*

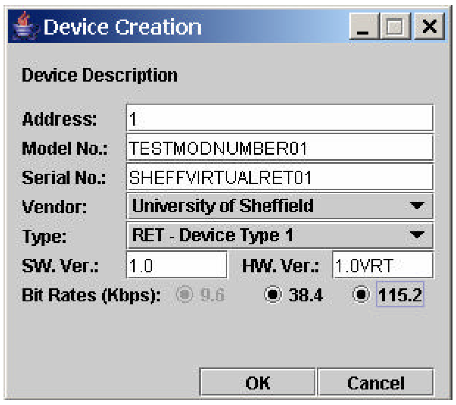
4. Virtual RET Device Test Interface

4.1. *Operation of Secondary Virtual Device*

In this mode, the command buttons remain disabled, since this application is emulating a ***Slave device***. Device creation can be performed by two procedures: either by loading a pre-configured device description file or by using the device creation button. This mode will allow only one RET/TMA device at a time. This means that if several devices are created using the Create Device Dialog or if a Device Description File with multiple devices is loaded, ***only the last device input*** is taken into account, the rest are ignored.

4.1.1. *Device creation: Create Button*

Opens the Device Creation Window. This provides a way of explicitly declaring a device. At least the Address, Serial number, vendor and device type must be specified here. Model number, software and hardware versions are part of the description data of a device, but not essential for communication.


Device Creation

Device Description

Address: 1
Model No.: TESTMODNUMBER01
Serial No.: SHEFFVIRTUALRET01
Vendor: University of Sheffield
Type: RET - Device Type 1
SW. Ver.: 1.0 HW. Ver.: 1.0VRT
Bit Rates (Kbps): ○ 9.6 ● 38.4 ● 115.2

OK Cancel

Figure 5. *Device Creation window*

The input fields are verified for format before being input:

- **Address**: integers 0 to 254. *If* "0" is input, a warning will be generated to inform the user that such address corresponds to a manufacturer default value. Unrecognised values/format will be defaulted to 0.
- **Model Number** (Product number): 15 ASCII characters, longer strings are truncated. If non-ASCII characters are found, they will be rejected and replaced by 0x00.
- **Serial Number**: 17 ASCII characters, longer strings are truncated. The input format verifiers provide no padding, if a shorter, non-zero length string is input, the serial number length will be equal to the length of the input string. If Non-ASCII characters are found, a message informs the user, if ignored, the input is truncated on the position where the first non-ASCII character is found.

- **Vendor**: List of 50 registered vendor codes up to release 1.0 of the specification.
- **Type**: 4 registered device types up to release 1.0. Devices Type 1 and 3 are Remote Electrical Tilt units, 2 is assigned to Tower Mounted Amplifiers and 4 to Tower Mounted Boosters.
- **SW/HW version**: ASCII strings. Length not limited, but dependent on real secondary device resources.
- **Supported Bit Rates**: A radio button group showing the available bit rates for an AISG compliant device. 9.6 KBPS is selected by default and can not be modified. 38.4 and 115.2 KBPS are optional. The software implements the standard bit-rate change procedure, which will be attempted after long periods of silence over the link.

Once information is input, clicking on the OK button creates an instance of Device for internal representation and sets the information input in the main screen interface.

4.1.2. Device creation: Load Button

Loads a Device Description File. This has been implemented with the primary objective of avoiding manual introduction of the information. A Device Description File is a text file where the device state is stored (basic information as well as communication status, alarm logs, current errors detected). ***Annex A*** presents the ***general device description file template*** with descriptions and comments.

Care should be taken when modifying a device description file, since incorrect values might render it useless and errors will be generated in the application. Please read carefully the comments in the provided template.

4.2. Save Button

At any point, the state of the device can be saved to a Device Description File. This allows the user to verify byte-to-byte the internal state of a device in order to analyse its behaviour.
In secondary mode, the application only saves the device on display. Even if more than one device was declared in the emulator, only the last active will be saved

4.3. Delete button.

Remains disabled in this mode, since whenever a device is no longer required, a new one can be loaded/created and the previous one is lost (if not saved). However, all devices declared should have different unique Identification numbers (unique ID is the combination of a vendor code with a serial number).

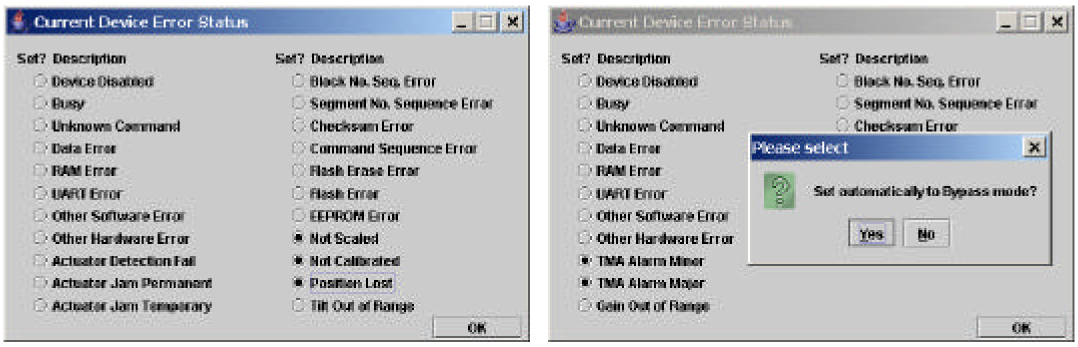
4.4. Alarms List Button.

This is an alarm history report. The number of entries in the log is limited to 35 to emulate the limited buffer in a real device.

4.5. Device Errors.

This button will open a dialog as the one shown in figure 6. It is possible to modify the error values, hence simulating error appearance/disappearance. This allows for tests on the alarm reporting mechanism in the primary station.

The TMA mode also asks to change mode in the secondary upon appearance of a major alarm. This change is allowed here, since in TMA devices the switch in modes should be performed upon appearance of a problem in the normal status.



Set?	Description	Set?	Description
○	Device Disabled	○	Block No. Seq. Error
○	Busy	○	Segment No. Sequence Error
○	Unknown Command	○	Checksum Error
○	Data Error	○	Command Sequence Error
○	RAM Error	○	Flash Erase Error
○	UART Error	○	Flash Error
○	Other Software Error	○	EEPROM Error
○	Other Hardware Error	●	Not Scaled
○	Actuator Detection Fail	●	Not Calibrated
○	Actuator Jam Permanent	●	Position Lost
○	Actuator Jam Temporary	○	Tilt Out of Range

Set?	Description	Set?	Description
○	Device Disabled	○	Block No. Seq. Error
○	Busy	○	Segment No. Sequence Error
○	Unknown Command	○	Checksum Error
○	Data Error		
○	RAM Error		
○	UART Error		
○	Other Software Error		
○	Other Hardware Error		
●	TMA Alarm Minor		
●	TMA Alarm Major		
○	Gain Out of Range		

Please select

Set automatically to Bypass mode?

Yes No

Figure 6. *Device error status. Modifiable screen in Virtual Secondary Device, left RET Screen, right TMA screen with message linked to TMA Major alarms.*

4.6. Communication Exchange Log

This section of the screen displays the frame exchange between the primary and the secondary station and shows the result of the incoming/outgoing frame integrity checks. The information shown in both areas is saved in a log file every 120 lines, after which the display buffers are cleared. The logs are stored in the path ***<installation root>\logFiles***. The files are named:

- *Secondary_SentCmds_YYYYMMDD.log* – Received commands and messages sent by the primary station.
- *Secondary_RcvdCmds_YYYYMMDD.log* – Responses sent to the primary station.

These files are saved also when closing the application and the result of the termination procedures of running threads and the application main thread is entered in the log.

An execution thread of the application handles log management. Each day that the application is run will have a pair of files associated to it. Since each log is kept as a plain text file containing the information exchange of the day, they usually consume a significant amount of hard drive space.

In order to save resources on the PC there is a maintenance task that obtains a list of the files in the system log directory and erases all files that are more than 7 days old. This task is run at start-up time and, if the application is left connected continuously, every day at midnight.

5. KNOWN BEHAVIOURS AND LIMITATIONS

The virtual Secondary device has many limitations, since it was developed only as a testing tool. Almost nothing in the tool emulates the behaviour of a TMA or RET system further than the communication protocol and information format. The behaviours linked to a real device would depend on sensors and actuators, which are not simulated here.

The tool responds to received commands, verifies the integrity and format and then changes its state accordingly and replies with the corresponding frame. However there are simplifications, such as the fact that no routines for "Self Test" or "Vendor Specific Commands" are implemented.

Another example of the test-only nature of the program is the fact that responses to Vendor Specific Commands are simply frames containing the bytes of the vendor code followed by the bytes "CA FE BA BE CA FE BA BE". The objective is to let the tester know that the format of a vendor specific command has been understood, but no further processing is performed by the secondary.

Most commands respond by default using the normal errorless response, except for commands where particular behaviours are tested. Write commands on a disabled device will produce FAIL response frames (***table 2***).

Clear Alarms	RET Calibrate*
Software Download*	RET SET Tilt*
Self Test	RET Send Configuration Data*
Set Device Data	TMA Set Mode*
Device Memory Write	TMA Set Gain*

Table 2. *Commands that require an Enabled device.*
** These "write" operations have not been implemented in this version of the emulator.*

Memory access is limited to a space of 2048 bytes, attempting an access beyond this limit will produce a FAIL response from the software. Verification of what has been written into memory can be done by verifying the frame exchange or by using a read memory command. No device memory view has been implemented for the secondary virtual device mode.

The free demo version of the software does not implement the full AISG command set. Please refer to appendix B for a summary of the implemented commands in the software.

ANNEX A. DEVICE DESCRIPTION FILE TEMPLATE

```
# Template for a device configuration file, comments are allowed only in the header of
# this file v1.1
#-------------------------------------------------------------------------------------
#  1 Device Logical Address – decimal number representation (0 - 254)
#  2 Device Type – decimal number (1 – 4)
#  3 Device Vendor Code index - integer, decimal (0 – 49)
#  4 Device Serial number - hex string format, 17-ASCII representation of the serial
#                        number
#  5 Device Hardware version – ASCII String, if not defined use '---'
#  6 Device Software version – ASCII String, if not defined use '---'
#  7 Connected Status - 'true' for connected, 'false' otherwise
#  8 Enabled Status - 'true' for enabled, 'false' otherwise
#  9 Supported Bit rates - hex format, maximum three bytes (0x00, 0x01, 0x02)
# 10 Current Bit Rate - decimal number use 0 (9.6Kbps) by default
# 11 Antenna Bearing - decimal number
# 12 Antenna Installed Mechanical Tilt - decimal number representation (deg/10, ie 3.2
#                                     deg -> 32)
# 13 Device Model number value - hex string format, 15-ASCII
#                             number
# 14 Device Installation date - hex string format, 6 bytes ASCII installation date
# 15 Device Installer ID - hex format, 5 bytes ASCII installer ID
# 16 Device Base Station ID - hex format, 12 bytes ASCII representation of the
#                              base station ID
# 17 Device Sector ID - hex format, 4 bytes ASCII representation of the sector ID
# 18 Device errors - Errors present at the time of storage, use '---' for no errors
# 19 Device alarm log - Alarm log at the time of storage, use '---' for an empty log
# 20 Device alarm log pointer - Pointer to the start of the next alarm report, always
#                            0 for primary files, or set to 0 in a secondary
#                            file to generate a report of the whole log on the next
#                            opportunity
# 21 Device send sequence number N(s) - 3 bits, decimal representation of the sequence
#                                       number send (HDLC modulo 8)
# 22 Device receive sequence number N(r) - 3 bits, decimal representation of the
#                                          sequence number receive (HDLC modulo 8)
# 23 Device has received reset command - if set to 'true', device will reset on the
#                                 next RR, I or RNR with the correct N(r)
#-------------------------------------------------------------------------------------
# If device is RET, then the following lines are interpreted as:
# 24 RET Device frequency bands - byte, decimal representation from Appendix D AISG1
# 25 RET Device Beam Width per frequency band - hex format, 3 bytes, one beamwidth per
#                                            band (degrees) if not all three are used,
#                                            pad with 0x00
# 26 RET Device Gain per frequency band - hex format, 3 bytes, one gain per band
#                                        (db/10, ie 10 dB -> 100) if not all three are
#                                        used, pad with 0x00
# 27 RET Antenna Model number - hex format, 15 bytes ASCII antenna model number
# 28 RET Antenna Serial number - hex format, 17 bytes ASCII antenna serial number
# 29 RET Configuration data - hex format, 70 bytes max. of configuration data
# 30 RET Maximum tilt - short, decimal representation (deg/10, ie 3.2 deg -> 32, -3.2
#                       deg -> -32)
# 31 RET Minimum tilt - short, decimal representation (deg/10, ie 3.2 deg -> 32, -3.2
#                       deg -> -32)
# 32 RET Current electrical tilt - short, decimal representation (deg/10, ie 3.2
#                                  deg -> 32, -3.2 deg -> -32)
#-------------------------------------------------------------------------------------
# If device is TMA, then the following lines are interpreted as:
# 24 TMA Model number - hex format, 15 bytes ASCII representation of the tma
#                    model number
# 25 TMA Serial number - hex format, 17 bytes ASCII representation of the tma
#                     serial number
# 26 TMA type – decimal, 0-3 based on appendix D of standard
# 27 TMA Reception Band – 4 bytes, formatted as appendix D:
#                      <Min. Low ><Min. High><Max. Low ><Max. High>
# 28 TMA Transmission Band – 4 bytes, formatted as appendix D:
#                         <Min. Low ><Min. High><Max. Low ><Max. High>
# 29 TMA Max. Gain – decimal 0 – 255 -> if 100 is input = 25.0 dB
# 30 TMA Min. Gain – decimal 0 – 255 -> if 100 is input = 25.0 dB
# 31 TMA Gain Resolution – decimal 0 – 255 -> if 100 is input = 6.25 dB
# 32 TMA Current gain value – decimal 0 – 255 -> if 100 is input = 25.0 dB
# 33 TMA Current Mode – 0: normal, 1: bypass
#-------------------------------------------------------------------------------------
1
1
44
53 48 45 46 46 56 49 52 54 55 41 4c 52 45 54 30 31
1.0VRT
1.0
false
false
00 01 02
0
120
254
54 45 53 54 4d 4f 44 4e 55 4d 42 45 52 30 31
32 33 30 32 30 34
74 73 74 30 31
73 68 65 66 66 69 65 6c 64 30 30 31
52 43 33 34
09
09 00 09 01
0
2
4
false
3
0c 09 00
15 0f 00
41 4e 54 45 4e 4e 41 4d 4f 44 4e 4f 30 30 31
41 4e 54 45 4e 4e 41 53 45 52 4e 4f 30 30 30 30 31
00 00 00 00 00 00 00 00 00
100
0
0
2
2
44
53 48 45 46 46 56 49 52 54 55 41 4c 54 4d 41 30 31
1.0VRT
1.0
false
false
00 01 02
0
120
254
54 45 53 54 4d 4f 44 4e 55 4d 42 45 52 30 31
32 33 30 32 30 34
54 53 54 30 31
53 48 45 46 46 49 45 4c 44 55 4e 49
52 43 33 34
09
09 00 09 01 09 00 09 01 09 00 09 01
0
1
7
false
54 4d 41 4d 4f 44 45 4c 4e 4f 30 30 30 30 31
54 4d 41 53 45 52 49 41 4c 4e 4f 30 30 30 30 30 31
3
ea 24 b2 25
98 21 f6 22
24
5
12
0
1
```

Note that at the end of the file there MUST NOT be a line feed.

ANNEX B. SUMMARY OF IMPLEMENTED COMMANDS IN THIS RELEASE

AISG Application Layer Command Set	***Global Commands***	*Mandatory*		GetDeviceType	***Implemented***
				Reset	***Implemented***
				GetErrorStatus	***Implemented***
				GetHWSWInfo	***Implemented***
				ClearAlarms	***Implemented***
				Alarm	***Implemented***
				EnableDevice	***Implemented***
				DisableDevice	***Implemented***
				SelfTest	***Implemented***
				ReadMemory	***Implemented***
				WriteMemory	***Implemented***
				GetSupportedBitRates	***Implemented***
				SetDeviceData	***Implemented***
				GetDevice Data	***Implemented***
		Optional	*Software Download Command Sequence*	DownloadStart	***Not available***
				StoreDataStart	***Not available***
				StoreDataBlock Segment	***Not available***
				DownloadEnd	***Not available***
				SetBitRate*	***Implemented***
	RET Specific commands	Calibrate	***Not available***		
		SendConfigData	***Not available***		
		SetTilt	***Not available***		
		GetTilt	***Not available***		
	TMA Specific commands	SetTMAMode	***Not available***		
		GetTMAMode	***Not available***		
		SetTMAGain	***Not available***		
		GetTMAGain	***Not available***		

** SetBitRate will be understood if the command issued from the primary station is embedded in an HDLC UI frame with the control byte set to **0x03** (UI with poll bit cleared). The secondary device runs a thread that verifies long periods of "silence" over the link.*